

```

BBBBBBBBBBBBBB      AAAAAAAAAA      SSSSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTTTT      LLL
BBBBBBBBBBBBBB      AAAAAAAAAA      SSSSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTTTT      LLL
BBBBBBBBBBBBBB      AAAAAAAAAA      SSSSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTTTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBBBBBBBBBBBBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBBBBBBBBBBBBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBBBBBBBBBBBBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAAAAAAAAAAAAAAAAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAAAAAAAAAAAAAAAAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAAAAAAAAAAAAAAAAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBBBBBBBBBBBBB      AAA      AAA      SSSSSSSSSSSSSS      RRR      RRR      TTT      TTT      LLL
BBBBBBBBBBBBBB      AAA      AAA      SSSSSSSSSSSSSS      RRR      RRR      TTT      TTT      LLL
BBBBBBBBBBBBBB      AAA      AAA      SSSSSSSSSSSSSS      RRR      RRR      TTT      TTT      LLL

```

```
BBBBBBBBB      AAAAAA      SSSSSSSSS  XX      XX      LL      AAAAAA      TTTTTTTTTT  EEEEEEEEEEE
BBBBBBBBB      AAAAAA      SSSSSSSSS  XX      XX      LL      AAAAAA      TTTTTTTTTT  EEEEEEEEEEE
BB      BB      AA      AA      SS      XX      XX      LL      AA      AA      TT      EE
BB      BB      AA      AA      SS      XX      XX      LL      AA      AA      TT      EE
BB      BB      AA      AA      SS      XX      XX      LL      AA      AA      TT      EE
BBBBBBBBB      AA      AA      SSSSSSS  XX      XX      LL      AA      AA      TT      EEEEEEEEE
BBBBBBBBB      AA      AA      SSSSSSS  XX      XX      LL      AA      AA      TT      EEEEEEEEE
BB      BB      AAAAAAAAAA      SS      XX      XX      LL      AAAAAAAAAA      TT      EE
BB      BB      AAAAAAAAAA      SS      XX      XX      LL      AAAAAAAAAA      TT      EE
BB      BB      AA      AA      SS      XX      XX      LL      AA      AA      TT      EE
BB      BB      AA      AA      SS      XX      XX      LL      AA      AA      TT      EE
BBBBBBBBB      AA      AA      SSSSSSSSS  XX      XX      LLLLLLLLLL  AA      AA      TT      EEEEEEEEE
BBBBBBBBB      AA      AA      SSSSSSSSS  XX      XX      LLLLLLLLLL  AA      AA      TT      EEEEEEEEE
```

```
LL      IIIIII      SSSSSSSSS
LL      IIIIII      SSSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSSS
LL      II      SSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIIII      SSSSSSSSS
LLLLLLLLLL  IIIIII      SSSSSSSSS
```

BASSXLATE  
Table of contents

(2)	49	DECLARATIONS	
(3)	92	BASSXLATE	- Perform BASIC XLATE function

```

0000 1      .TITLE BASXLATE
0000 2      .IDENT /1-004/
0000 3
0000 4
0000 5      ; File: BASXLATE.MAR EDIT: RNH1004
0000 6
0000 7      *****
0000 8      *
0000 9      * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 10     * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 11     * ALL RIGHTS RESERVED.
0000 12     *
0000 13     * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 14     * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 15     * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 16     * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 17     * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 18     * TRANSFERRED.
0000 19     *
0000 20     * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 21     * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 22     * CORPORATION.
0000 23     *
0000 24     * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 25     * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 26     *
0000 27     *
0000 28     *
0000 29     *++
0000 30     * FACILITY: BASIC code support
0000 31     *
0000 32     * ABSTRACT:
0000 33     *
0000 34     *      This module implements the BASIC-PLUS-2 XLATE function.
0000 35     *
0000 36     * ENVIRONMENT: User Mode, AST Reentrant
0000 37     *
0000 38     *--
0000 39     * AUTHOR: R. WILL, CREATION DATE: 18-May-79
0000 40     *
0000 41     * MODIFIED BY:
0000 42     *
0000 43     * R. Will, : VERSION 1
0000 44     * 1-001 - Original
0000 45     * 1-002 - Change calls to STR$COPY. JBS 16-JUL-1979
0000 46     * 1-003 - Change a INCW R1 to INCL R1. R1 contains an address. FM 5-FEB-81
0000 47     * 1-004 - Change shared external references to G^ RNH 25-Sep-81

```



## DECLARATIONS

```
0000 49      .SBTTL  DECLARATIONS
0000 50      :
0000 51      : INCLUDE FILES:
0000 52      :
0000 53      :
0000 54      $DSCDEF      ; define descriptor offsets
0000 55      :
0000 56      :
0000 57      : EXTERNAL DECLARATIONS:
0000 58      :
0000 59      .DSABL  GBL      ; Prevent undeclared
0000 60      :              ; symbols from being
0000 61      :              ; automatically global.
0000 62      :
0000 63      .EXTRN  STR$COPY_DX_R8      ; copy input string to temp
0000 64      :              ; and temp string to output
0000 65      .EXTRN  STR$COPY_R_R8      ; copy temp str to dest str
0000 66      .EXTRN  STR$GET1_DX      ; allocate temp string
0000 67      .EXTRN  STR$FREE_DX      ; deallocate temp string
0000 68      .EXTRN  LIB$GET_VM      ; allocate heap memory
0000 69      .EXTRN  LIB$FREE_VM      ; deallocate heap memory
0000 70      :
0000 71      :
0000 72      : MACROS:
0000 73      :
0000 74      :
0000 75      :
0000 76      : EQUATED SYMBOLS:
0000 77      :
0000 78      :
00000000 0000 79      null = ^X00
0000 80      :
0000 81      :
0000 82      : OWN STORAGE:
0000 83      :
0000 84      :
0000 85      :
0000 86      : PSECT DECLARATIONS:
0000 87      :
00000000 88      .PSECT _BAS$CODE PIC, USR, CON, REL, LCL, SHR, -
0000 89      EXE, RD, NOWRT, LONG
0000 90
```

BASSXLATE - Perform BASIC XLATE functi  
          92 .SBTTL BASSXLATE - Perform BASIC XLATE function

0000 93 :  
0000 94 :  
0000 95 :  
0000 96 :  
0000 97 :  
0000 98 :  
0000 99 :  
0000 100 :  
0000 101 :  
0000 102 :  
0000 103 :  
0000 104 :  
0000 105 :  
0000 106 :  
0000 107 :  
0000 108 :  
0000 109 :  
0000 110 :  
0000 111 :  
0000 112 :  
0000 113 :  
0000 114 :  
0000 115 :  
0000 116 :  
0000 117 :  
0000 118 :  
0000 119 :  
0000 120 :  
0000 121 :  
0000 122 :  
0000 123 :  
0000 124 :  
0000 125 :  
0000 126 :  
0000 127 :  
0000 128 :  
0000 129 :  
0000 130 :  
0000 131 :  
0000 132 :  
0000 133 :  
0000 134 :  
0000 135 :  
0000 136 :  
0000 137 :  
0000 138 :  
0000 139 :  
0000 140 :  
0000 141 :  
0000 142 :  
0000 143 :  
0000 144 :  
0000 145 :  
0000 146 :  
0000 147 :  
41FC 0000 148

++  
FUNCTIONAL DESCRIPTION:

This routine implements the BASIC-PLUS-2 XLATE function.  
For AST re-entrancy, the routine will create a local dynamic string  
descriptor and call STR\$COPY to copy the source string to the local  
(instead of using any mechanism to prevent AST level routines  
from writing to the source string and moving it from under us).  
The routine will also create a local dynamic string descriptor and  
call allocate to get a string to translate into. The routine will  
then use both local strings (which will not need to get larger) to do  
the translating. The routine will use the MOVTUC to translate until  
the translated character is the NULL character. The NULL will not be  
written to the destination string, and the translation will continue  
with the next character. After the translating is finished,  
the routine will call STR\$COPY to copy the edited string to the  
destination string.

## CALLING SEQUENCE:

CALL BASSXLATE (dest\_string.wx.dx, src\_string.rx.dx, table.rx.dx)

## INPUT PARAMETERS:

src\_string = 8  
table = 12

## IMPLICIT INPUTS:

NONE

## OUTPUT PARAMETERS:

dest\_string = 4

## IMPLICIT OUTPUTS:

NONE

## FUNCTION VALUE:

## COMPLETION CODES:

NONE

## SIDE EFFECTS:

This routine calls STR\$COPY and STR\$FREE1 and therefore will  
allocate dynamic string space to a temporary, may allocate dynamic  
string space to the destination string, and may cause any of the  
their error messages to be signalled. This routine also calls  
LIB\$GET\_VM and LIB\$FREE\_VM and so any of their errors may be  
signalled.

--  
.ENTRY BASSXLATE, ^M<R2,R3,R4,R5,R6,R7,R8,IV>

BASXLATE

- Perform BASIC XLATE functi

```
0002 149
0002 150 :+
0002 151 : Create a local descriptor and copy the input string to it using STR$COPY
0002 152 :-
0002 153
    51 08 AC D0 0002 154      MOVL    src_string(AP), R1      ; pointer to src string
    7E D4 0006 155      CLRL    -(SP)                      ; address of local string
020E0000 BF DD 0008 156      PUSHL   #<<DSC$K_CLASS_D @ 24> ! <DSC$K_DTYPE_T @ 16>> ; fill type, class and length
    50 5E D0 000E 157      MOVL    SP, R0                  ; R0 points to local descriptor
00000000 GF 16 0011 158      JSB     G^STR$COPY_DX_R8        ; copy string to local
    0017 160
    0017 161 :+
    0017 162 : Create a local descriptor and allocate space to it, to use as destination
    0017 163 : string for MOVTUC
    0017 164 :-
    0017 165
    020E0000 7E D4 0017 166      CLRL    -(SP)                      ; address of local string
    BF DD 0019 167      PUSHL   #<<DSC$K_CLASS_D @ 24> ! <DSC$K_DTYPE_T @ 16>> ; fill type, class and len
    5E DD 001F 168      PUSHL   SP                          ; point to descriptor
    08 BC 3F 0021 169      PUSHAW  @src_string(AP)          ; length to allocate
00000000 GF 02 FB 0024 170      CALLS   #2, G^STR$GET1_DX      ; allocate space
    002B 171
    002B 172
    002B 173 :+
    002B 174 : Call LIB$GET_VM to allocate 256 bytes to use for translate table
    002B 175 : and create the translation table
    002B 176 :-
    002B 177
    7E 00000100 7E D4 002B 178      CLRL    -(SP)                      ; space for memory pointer
    BF DD 002D 179      MOVL    #256, -(SP)                  ; # bytes to allocate
    04 AE DF 0034 180      PUSHAL  4(SP)                      ; ptr to output parameter
    04 AE DF 0037 181      PUSHAL  4(SP)                      ; ptr to byte count
    00000000 GF 02 FB 003A 182      CALLS   #2, G^LIB$GET_VM    ; allocate the space
    50 0C BC 7D 0041 183      MOVQ    @table(AP), R0          ; get table pointer and length
04 BE 0100 BF 00 61 50 2C 0045 184      MOVCS   R0, (R1), #null, #256, @4(SP) ; fill the translate table
    004E 185
    004E 186 :+
    004E 187 : fill registers for the MOVTUC loop
    004E 188 : R0 src len
    004E 189 : R1 src pointer
    004E 190 : R3 address of translation table
    004E 191 : R4 dest len
    004E 192 : R5 dest pointer
    004E 193 :-
    004E 194
    50 10 AE 7D 004E 195      MOVQ    16(SP), R0              ; R0 & R1 <- len & ptr for src
    54 08 AE 7D 0052 196      MOVQ    8(SP), R4               ; R4&R5 <- len & ptr for dest
    53 04 AE D0 0056 197      MOVL    4(SP), R3              ; R3 has addr of extendd table
    005A 198
    005A 199 :+
    005A 200 : Registers are initialized, so MOVTUC until get a NULL, increment src ptr
    005A 201 : decrement src len to describe string remaining after NULL translation.
    005A 202 : Then continue translating.
    005A 203 :-
    005A 204
    65 54 63 00 61 50 2F 005A 205 1$: MOVTUC R0, (R1), #null, (R3), R4, (R5) ; find null translation
```



```

BASSXLATE
1-004
206      TSTW      R0
207      BEQLU     FINISH
208      DECW      R0
209      INCL      R1
210      BRB       1$
211
212      ;+
213      ; The string has been translated. Free the VM used for the translate table.
214      ; Copy the temporary storage to the destination string. (Note that the trans-
215      ; lated length is the source length minus the number of unfilled bytes in the
216      ; temporary string left in R4 by the MOVTUC.) Deallocate the temporary string
217      ; and the copied source string. Clean up the stack and return.
218      ; -
219
220      FINISH:
221      PUSHAL     4(SP)
222      PUSHAL     4(SP)
223      CALLS      #2, G^LIB$FREE_VM
224      CLRQ       (SP)+
225
226      SUBW3      R4, @src_string(AP), R1
227      MOVL       4(SP), R2
228      MOVL       dest_string(AP), R0
229      JSB        G^STR$COPY_R_R8
230
231      PUSHAL     (SP)
232      CALLS      #1, G^STR$FREE1_DX
233      CLRQ       (SP)+
234
235      PUSHAL     (SP)
236      CALLS      #1, G^STR$FREE1_DX
237      CLRQ       (SP)+
238
239      RET
240
241      .END
242
243      ; End of BASSXLATE

```



BASSXLATE  
Symbol table

L 7

16-SEP-1984 00:01:59 VAX/VMS Macro V04-00  
6-SEP-1984 10:40:27 [BASRTL.SRC]BASSXLATE.MAR;1

Page 6  
(3)

BASSXLATE	= 00000000	RG	02
DEST_STRING	= 00000004		
DSC&R_CLASS_D	= 00000002		
DSC&K_DTYPE_T	= 0000000E		
FINISH	0000006B	R	02
LIB\$FREE_VM	*****	X	00
LIB\$GET_VM	*****	X	00
NULL	= 00000000		
SRC_STRING	= 00000008		
STR\$COPY_DX_R8	*****	X	00
STR\$COPY_R_R8	*****	X	00
STR\$FREE_T_DX	*****	X	00
STR\$GET1_DX	*****	X	00
TABLE	= 0000000C		

-----  
! Psect synopsis !  
-----

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
_BASSCODE	000000A4 ( 164.)	02 ( 2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

-----  
! Performance indicators !  
-----

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.08	00:00:00.65
Command processing	117	00:00:00.43	00:00:02.18
Pass 1	137	00:00:01.84	00:00:04.30
Symbol table sort	0	00:00:00.17	00:00:00.31
Pass 2	56	00:00:00.61	00:00:01.58
Symbol table output	3	00:00:00.02	00:00:00.02
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	348	00:00:03.17	00:00:09.06

The working set limit was 1050 pages.  
9144 bytes (18 pages) of virtual memory were used to buffer the intermediate code.  
There were 10 pages of symbol table space allocated to hold 143 non-local and 1 local symbols.  
241 source lines were read in Pass 1, producing 13 object records in Pass 2.  
8 pages of virtual memory were used to define 7 macros.

-----  
! Macro library statistics !  
-----

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4

190 GETS were required to define 4 macros.

BASXLATE  
VAX-11 Macro Run Statistics

M 7

16-SEP-1984 00:01:59 VAX/VMS Macro V04-00  
6-SEP-1984 10:40:27 [BASRTL.SRC]BASXLATE.MAR;1

Page 7  
(3)

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS\$:BASXLATE/OBJ=OBJ\$:BASXLATE MSRC\$:BASXLATE/UPDATE=(ENH\$:BASXLATE)



0034 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY